SEGURANÇA EM SISTEMAS COMPUTACIONAIS – DESENVOLVIMENTO DE UM MECANISMO DISTRIBUÍDO PARA DETECÇÃO DE INTRUSÕES EM REDES TCP/IP

André Luís Fávero[1]
Cristofer Veloso[2]
Iderlei Rodrigo Merlugo[2]
Marco Antônio Sandini Trentin[3]
Universidade de Passo Fundo
Curso de Ciência da Computação – ICEG
Passo Fundo – RS – Brasil
E-mail: {andre, soft, iderlei, trentin}@inf.upf.tche.br

RESUMO

Visto o grande avanço que a informática vem proporcionando, junto dele aumenta cada vez mais a possibilidade de acontecerem falhas e estas provocarem prejuízos incalculáveis. Por este motivo o investimento na área segurança de informações deve ganhar uma atenção maior, pois tem-se a necessidade de desenvolver mecanismos e tecnologias que tenham como objetivo prevenir de acontecimentos indesejáveis. Este documento apresenta um contexto sobre ferramentas para detecção de intrusão, sugerindo o desenvolvimento de um aplicativo que atue nesta área de uma forma distribuída.

PALAVRAS-CHAVE: Segurança, Intrusões, Sistema de Detecção a Intrusões.

ABSTRACT

Seen the great advance that computer science comes providing, together of it more magnifies each time the possibility to happen fails and these fails provoke incalculable damages. For this reason the investment in the area security of information must gain an bigger attention, therefore it is had necessity to develop mechanisms and technologies that have as objective the prevention of events undesirables. This document presents a context of intrusion detention tools, suggesting the development of a application one that it acts in this area in a distributed form.

KEY-WORDS: Security, Intrusion, Intrusion Detection System.

1 Aluno de graduação do curso de Ciência da Computação da UPF, bolsista PIBIC/UPF.
2 Aluno de graduação do curso de Ciência da Computação da UPF.
3 Professor do curso de Ciência da Computação da UPF, aluno de doutorado do PGIE da UFRGS.

1 INTRODUÇÃO

Apesar de haver um enorme crescimento na utilização de sistemas de informática em toda e qualquer empresa, estas estando ligadas à Internet ou não, o investimento na segurança desses sistemas não vem crescendo na mesma proporção. Preocupadas em rapidamente informatizar seus negócios, as empresas não se mobilizam e deixam de exigir de seus fornecedores de tecnologia um mínimo de padrão de segurança. Em prol do desempenho, diversas atitudes simples, deixam de ser feitas, fazendo com que aparecem brechas de segurança nos sistemas, e estas vindo a ser exploradas.

Esta falta de segurança, muitas vezes causada por falta de atenção a detalhes técnicos. Isto faz com que surjam motivos que na maioria da vezes estão relacionados com descontentamentos, busca de auto afirmação, *status*, adrenalina, e principalmente por inconseqüência, para que possíveis usuários venham a conseguir quebras de segurança, sendo levados a tirar proveito destas situações.

Tem-se então ambientes ricos para atitudes que venham a provocar invasões e então quebras de segurança, que são facilitadas pela popularização da Internet, onde se consegue de maneira rápida e fácil ferramentas que tem por objetivo explorar falhas e dar acesso ao sistemas, caracterizando um uso indevido do sistemas.

2 INTRUSÕES

De forma simples pode-se dizer que um intruso é alguém que tenta invadir um sistema ou fazer mau uso do mesmo. Discute-se então uma forma para definir um invasão, ou seja, o que é tentar invadir um sistema, ou o que é fazer mau uso do mesmo. A forma para diferenciar, então, as ações legítimas das ações nocivas, é aplicada através da definição de uma política de segurança, pois a menos que se tenha conhecimento do que é proibido no sistema, não tem sentido tentar "travar" intrusões.

Na maioria das vezes uma intrusão é caracterizada como uma tentativa de comprometer a integridade, a confidencialidade ou a disponibilidade de um recurso. Surge então a necessidade de um sistema que detecte estas invasões, *Intrusion Detection System, (IDS). Intrusion Detection* é a prática de usar ferramentas de maneira automatizada, tendo como um dos objetivos principais detectar intrusões em tempo real. Esta ferramenta é executada de forma constante e somente gera uma notificação quando detecta algo que seja suspeito ou ilegal. Trabalha de maneira parecida a um sensor, o qual deverá identificar padrões de ataques conhecidos previamente, e reportá-los ao administrador do sistema, se estes vierem a acontecer. Este tipo é caracterizado como sistema baseado em regras. Poderá ser também um sistema de detecção adaptável; este através de inteligência artificial aprenderá com o ambiente, passando a identificar novos padrões de intrusão conforme a definição da política de segurança.

3 DETECTANDO INTRUSÕES

Existem duas maneiras de se detectar uma possível quebra de segurança(intrusão). Uma através de anomalias, a qual procura medir *stats*, como a utilização do processador, atividade do disco, inícios de sessão do usuário, atividade em arquivos e assim por diante, gerando um certo padrão. Quando houver desvios deste padrão, o sistema enviará um sinal indicando que há um

desvio. O benefício de se fazer este tipo de comparação é que pode-se detectar anomalias sem ter que compreender a causa das mesmas.

Outra maneira de se detectar intrusões é através da identificação de assinaturas. Consiste em examinar o tráfego do segmento de rede a procura de padrões de ataque conhecidos, o que exige que cada padrão seja antes codificado no sistema. A maioria dos IDS comerciais são baseados nesta técnica. Estas duas categorias de IDS são descritas como:

- ✓ Network Intrusion Detection Systems (NIDS):

Monitoram os pacotes e os analisam de forma a descobrir se correspondem à uma tentativa de *scan* ou ataque. Rodam em um equipamento dedicado, devido a alta carga de processamento, pois a atuação se dá através da captura de todos pacotes que passam pelo segmento de rede no qual a máquina esta conectada. Tem como vantagem a possibilidade de monitorar várias máquinas ao mesmo tempo, ou mesmo o tráfego que passe pela rede ou *link* a qual o mecanismo encontra-se instalado.

- ✓ Host Intrusion Detection Tool:

Ferramenta instalada em um servidor ou desktop. Tem como função verificar periodicamente informações do sistema, por exemplo, *logs* e arquivos de sistemas, objetivando verificar se ocorreu alguma atividade ou alteração não autorizada. Atuam unicamente no equipamento, o qual está instalado.

Dentro desta as funcionalidades são descritas em duas categorias:

- ✓ System Integrity Verifiers (SIV):

Estes verificam os arquivos de sistema para identificar possíveis alterações, como exemplo um intruso alterando ou instalando um *backdoor*. Dentro desta categoria a ferramenta mais conhecida é a *TRIPWARE*, desenvolvida pela Purdue Research Foundation.

- ✓ Log File Monitors (LFM):

Analisam os arquivos de *logs* gerados pelo sistema, ou aplicação, procurando por padrões identificados como intrusão. Um exemplo deste tipo de ferramenta é o projeto *ABACUS*. Este é formado por três *softwares*, chamados de *portsentry, logcheck e hostcheck*. Cada um tem características próprias como identificação de *port scans* (tentativa de obter informações sobre os serviços - portas - abertos na máquina), envio de mensagem de alertas, analise do arquivos de mensagens do sistema, etc.[9]

3.1 Reconhecimento de Padrões

Padrões são segmentos de dados, contidos em um ou mais pacotes, os quais identificam uma intrusão, por exemplo:

- uma conexão para a porta 79 usando TCP contendo a string "root" é uma tentativa de buscar informações do usuário root da máquina;
- uma conexão para a porta 143 usando TCP contendo a string " e8c0 ffff ff /bin/sh" é uma tentativa explorar um bug conhecido do IMAP (*Internet Message Access Protocol*).

3.2 O que fazer quando o IDS reconhece o padrão de ataque

A reação pode ser somente informativa, mandando uma mensagem ao administrador, ou a execução de alguma tarefa, a qual pode ser: fechamento da conexão, bloqueio do pacotes vindos daquela origem ou, até muito pouco recomendado, um contra-ataque.

Existe, porém, um problema, que é a possibilidade dos alarmes serem falsos, pois em um momento a ocorrência do padrão pode não ser uma intrusão. Por isso, reações automáticas nem sempre são recomendadas.

4 FERRAMENTAS

Parte deste trabalho envolveu a busca e análise de *softwares* de detecção à intrusão, no intuito de conhecer o que existe no mercado, procurando por uma ferramenta que satisfizesse ao máximo as necessidades de segurança de uma rede.

Dentre as ferramentas analisadas, a que mais se destaca, chama-se SNORT. Escrita por Martin Roesch, sob licença GNU (*General Public License*) editada pela *Free Software Foundation.*

Por ser GNU, ou seja, livre para uso e alterações, e por ser uma ferramenta que possui muitos recursos, é estável e de funcionamento confiável, decidiu-se utilizá-la como base para a implementação de um DIDS (*Distribution Intrusion Detection System)* ou SDDI (*Sistema de Detecção Distribuída de Invasão).*

4.1 Descrição do SNORT

É um filtro de pacotes e *sniffer* baseado na *libpcap*. Atua como um analisador de protocolos, procurando por combinações de dados que sejam utilizados em ataques do tipo *buffer overflows, stealth port scans*, ataques CGI, tentativas de obtenção de OS *fingerprinting,* e também outras formas de intrusão. É considerado rápido por não necessitar de muitos recursos do sistema e também confiável[2].

Usa uma linguagem baseada em regras para descrever o que será caracterizado como violação.

Quanto ao mecanismo de alerta, é em tempo real, incorporando mecanismos como logs no sistema, logs em arquivos específicos, UNIX *socket*, ou mensagens *WinPopup* para clientes Windows usando Samba.

Possui *plugins* os quais permitem gravar logs em banco de dados, detecção em pequenos fragmentos, detecção de *portscan*[4].

5 PROPOSTA GERAL

O grupo de pesquisa tem como proposta a construção de uma ferramenta distribuída de IDS, que possa atuar como IDS em vários segmentos de rede. A ferramenta deve ser distribuída, sendo que foi escolhido o modelo agentes distribuídos, com um servidor (monitor) centralizado que irá manter os dados sobre o que os agentes devem procurar e mostrar o que foi encontrado por estes.

Nos clientes será usada a política de obtenção de informações usada na ferramenta SNORT, ou seja, este atuando em cada segmento de rede como agente.

No monitor, podendo ser instalado no *gateway* da rede ou em qualquer um dos segmentos (estes ligados entre si), ou ainda, em uma máquina qualquer na Internet, que consiga trocar informações com o monitor (*Figura 1*), ficarão armazenados os padrões definidos como violações e, também, os possíveis alertas gerados pelos agentes. As trocas de informações entre monitor e agente serão através de conexões *sockets.*

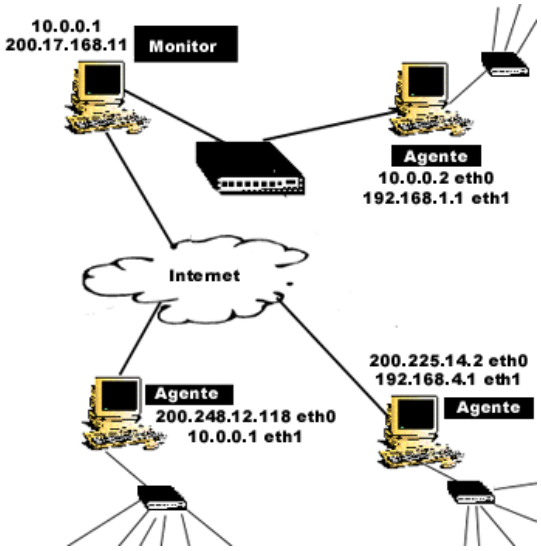

Figura 1: Modelo da ferramenta IDS proposta.

5.1 Monitor

A máquina monitor será o centro do IDS. Todos os padrões de ataque estarão armazenados em um banco de dados, provavelmente PostgresSQL, por ser de boa qualidade e *freeware.*

Os padrões serão repassados aos agentes toda vez que forem atualizados. Também neste banco de dados serão armazenados todos os relatórios emitidos pelos agentes, os quais irão conter um histórico de possíveis tentativas de intrusões. Estas serão apresentadas em uma página Web toda vez que o administrador fizer consultas a base de dados (*Figura 2*).

Figura 2: Gerenciamento do Monitor

Foi escolhida com interface de controle do monitor *browser*, visto que isto possibilita um gerenciamento remoto do IDS. Está interface será construída utilizando a linguagem PHP, a qual fará acesso a base de dados e a módulos escritos na linguagem C, os quais serão *sockets*, estes fazendo a comunicação para troca de informações entre o monitor e os agentes.

5.2 Agentes

Nos agentes serão implantados *daemons*, os quais terão por finalidade receber conexões *sockets* do monitor para atualizações dos padrões. Serão responsáveis também para conectar-se com o monitor e gravar logs de alertas na base de dados do mesmo.

Este *deamon* irá controlar todas a ações possíveis do processo que fará a captura dos pacotes.

A captura de pacotes será feita utilizando a biblioteca *libpcap* que implementa o *BPF* (*BSD Packet Filter)*, que é a capacidade de permitir o uso de filtros em capturas, levando em consideração argumentos que são campos do pacote TCP/IP.

Como comentado anteriormente, a estrutura do Snort será utilizada para a construção do agente, visto que o mesmo já possui um funcionamento bem estável e seria de certa maneira recriar algo que já existe e funciona corretamente.

O funcionamento do Snort se dá de maneira parecida ao *TcpDump*, sendo mais voltado a aplicações de segurança, isto porque o Snort analisa os dados de cada pacote. A camada de aplicação do pacote TCP/IP é decodificada, em busca de dados específicos, os quais são os padrões de ataque, nesta camada, permitindo então detectar atividades suspeitas de quebra de segurança. Toda arquitetura está voltada pela simplicidade, flexibilidade e boa performance na coleta e análise dos pacotes.

O agente de detecção atuará em etapas, sendo elas a captura de pacotes através do uso promíscuo da camada de hardware, após esta será efetuada a decodificação dos pacotes para busca dos padrões, e se estes forem encontrados, o processo irá enviar uma mensagem que conterá um alerta de detecção de violação ao *daemon*, que por sua vez irá reporta-lá ao monitor, onde será armazenada na base de dados (*Figura 3*).

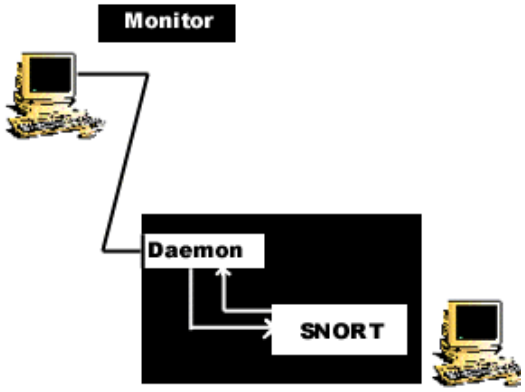

Figura 3: Troca de mensagens entre agente e monitor

A decodificação dos pacotes é organizada dependendo da pilha de protocolo de transporte de dados utilizada e de acordo com as definições do protocolo TCP/IP. Cada rotina aplicada desencapsula um pacote obtendo os dados do mesmo e montando-os numa estrutura.

A procura de padrões depende de regras as quais são mantidas em listas encadeadas bidimensionais que são chamadas de *Chain Headers* e *Chain Options*. O *Chain Headers* possui informações comuns a um tipo único de serviço, este podendo ser compartilhado por inumeras regras, por exemplo a origem e o destino da porta. Agregados a este estarão os *Chains Options*, possuindo características próprias a cada padrão.

Dependendo dos *Chain Headers*, existem uma lista de atributos comuns (*Chain Options*) ao *Chain Header*, isto é, com intuito de acelerar o processo de detecção de padrões, essas associações são condensadas em um único *Chain Header*, e então cada padrão possuirá um *Chain Options* específico (*Figura 4*).

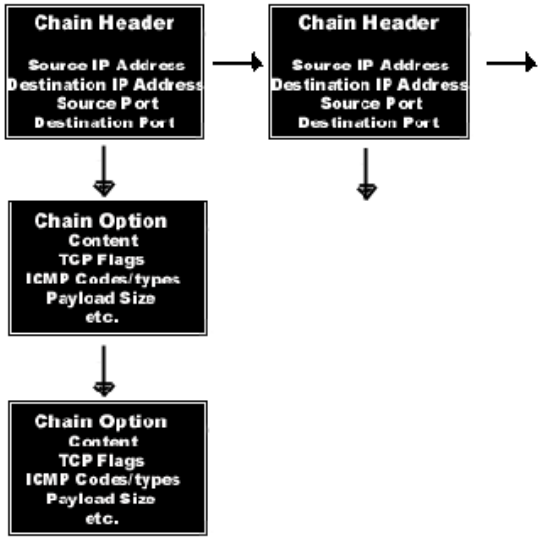

Figura 4: Chain Headers e Chain Options

Esta estrutura de regras (*Chains*) será procurada em cada pacote capturado. Só serão comparados os *Chains Options* cujos *Chain Headers* estiver ativos no sistema.

O sistema de alerta (já descrito anteriormente) entra em operação quando um padrão é reconhecido.

6 DEFINIÇÕES DE PADRÕES

A base de padrões ficará armazenada no monitor. O monitor se comunicará com o *deamon* dos agentes e transmitirá para estes todos os padrões catalogados, para um arquivo de configuração dos agentes.

6.1 Catalogando Padrões

Os padrões são defenidos através de regras, as quais podem ser simples e mesmo assim suficientes para detectarem um grande número de possíveis tentativas de intrusão. Ao escrever uma regra deve-se ter o cuidado para que esta não produza alarmes falsos em demasia.

Estas regras são formadas através de parâmetros os quais dizem respeito a protocolos, a endereços de origem para portas e máquinas, a endereços de destino para portas e máquinas e principalmente conteúdo dos pacotes, entre outros.

A regra abaixo verifica se algum dos pacotes que trafegam na rede, vindos de qualquer origem com destinos a *hosts* da rede 192.168.1, para o porta que referencia o serviço IMAP, contendo entre os dados a seguinte string, "|E8C0 FFFF FF|/bin/sh", produzirá um alerta indicando uma tentativa de explorar um bug do serviço.

alert tcp any any -> 192.168.1.0/24 143 (content:"|E8C0 FFFF FF|/bin/sh"; msg:"New IMAP Buffer Overflow detected!";)

7 CONCLUSÃO

Atualmente o projeto encontra-se em fase de estudo das implementações. Pretende-se no entanto o desenvolvimento da ferramenta em sua totalidade. Posteriormente a conclusão, a mesma será instalada e avaliada em laboratórios de informática da universidade, visto que este é um ambiente acadêmico onde o contexto é bastante favorável a violações, sendo ideal para testar suas funcionalidades.

Será distribuída nos termos da licença GNU, objetivando também uma continuação, melhora e aprimoramento da mesma.

Enfim, acreditamos que esta ferramenta será de grande valia, pois pecebe-se o crescimento do número de LAN's, e servidores de informações em empresas ou em outras instituições. Em vista disso, um ferramenta, como a proposta neste artigo, seria de grande valia.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] ANONYMOUS, *Maximum Security - A Hacker´s Guide to Protecting your Internet Site and Network,* 1997.

[2] ANONYMOUS, *Maximum Linux Security, A Hacker´s Guide to Protecting your Linux Server and Workstation,* Setembro de 1999.

[3] BACE, Rebecca Gurley, *Intrusion Detection*, 12/1999.

[4] SNORT, *The Lightweight Network Intrusion Detection System,*
URL: http://www.snort.org/ , 07/17/00

[5] CROSBIE, Mark, SPAFFORD, Gene, *Active Defence of a Computer System using Autonomous Agents.* COAST Group Dept. Of Computer Sciences Purdue University, Feb 1995.

[6] TCPDUMP Network Research Group of the Lawrence Berkeley National Laboratory, University of California, Berkeley, CA.
URL: ftp://ftp.ee.lbl.gov/tcpdump.tar.Z

[7] JACOBSON, C., LERES, C., MACCANNE, S. Lawrence Berkley National Laboratory, University of California, Berkley, CA.
URL: ftp://ftp.ftp.ee.lbl.gov/libpcap.tar.Z

[8] ABACUS PROJECT, *The Intrusion Prevention System*, Maio 2000.
URL: http://www.psionic.com/abacus/

[9] SECURITY MAGAZINE, Revista de Segurança em Informática, *Montando sua Infra-estrutura de Segurança de Informações,* Abril/Maio 2000.